

103-24-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of Howard Pfeffer and John Leddy

Serial No.: Not yet assigned

Group Art Unit:

Filed: Herewith

Examiner:

For: REDUCTION OF NETWORK SERVER LOADING

Assistant Commissioner for Patents
Box PATENT APPLICATION
Washington, D.C. 20231

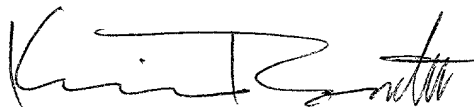
Dear Sir:

Enclosed please find the following:

1. Specification, abstract and claims (7 independent, 24 dependent, 31 total) (27 pages);
2. Informal drawings (7 figures, 5 sheets);
3. Declaration and Power of Attorney;
4. Assignment with Recordation Sheet;
5. One check in the amount of \$1,240.00 (\$690.00 for filing the application, \$312.00 for excess independent claims, \$198.00 for excess claims, and \$40.00 for filing the assignment); and,
6. Certificate of Express mailing.

The Commissioner is hereby authorized to charge any fee deficiency, or credit any overpayment, to Deposit Account No. 18-1579. The Commissioner is also authorized to charge Deposit Account No. 18-1579 for any future fees connected in any way to this application. Two copies of this letter are enclosed.

Respectfully submitted,



Kevin L. Pontius
Registration No. 37,512
Roberts Abokhair & Mardula, LLC
11800 Sunrise Valley Drive, Suite 1000
Reston, VA 20191-5302
(703) 391-2900

March 23, 2000

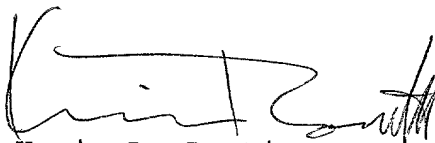
Atty. Docket No. 2585-001

CERTIFICATE OF EXPRESS MAILING

Express Mail Mailing Label Number EL528296193US

Date of Deposit: March 23, 2000

I hereby certify that the patent application of Harold Pfeffer and John Leddy for a **REDUCTION OF NETWORK SERVER LOADING** including the specification, abstract, and claims (7 independent, 24 dependent, 31 total) (27 pages); informal drawings (7 figures, 5 sheets); declaration and power of attorney; an assignment together with a recordation cover sheet; and a check in the amount of \$1,240.00 (\$690.00 for filing the application, \$312.00 for excess independent claims, \$198.00 for excess claims, and \$40.00 for filing the assignment), are being deposited with the United States Postal Service for "Express Mail" service under 37 C.F.R. § 1.10 on the date indicated above and are addressed to the Assistant Commissioner for Patents, Box Patent Application, Washington, D.C. 20231.



Kevin L. Pontius
Registration No. 37,512
Roberts Abokhair & Mardula, LLC
11800 Sunrise Valley Drive, Suite 1000
Reston, VA 20191-5302
(703) 391-2900

March 23, 2000

Atty. Docket No. 2585-001

1 Title: **REDUCTION OF NETWORK SERVER LOADING**

2 Inventors: **Howard Pfeffer and John Leddy**

3
4 **BACKGROUND OF THE INVENTION**

5 **1. Field of the Invention**

6 The present invention relates generally to the field of data networks. More
7 particularly, the present invention relates to reduction of traffic handling load on network
8 servers by decentralization of mail handling protocols.

9 **2. Background Information**

10 Over the last twenty years, the demand for data network services has grown
11 rapidly. Many large networks have been built by a number of providers to meet the
12 voracious demand for bandwidth to handle data traffic.

13 Data networks are commonly used for *inter alia* transmission of electronic mail
14 messages (hereinafter e-mail). According to Post Office Protocol (hereinafter POP)
15 standard for e-mail handling, transmitted e-mail messages are routed to a centralized mail
16 server facility. Those e-mail messages are warehoused at the centralized mail server until
17 retrieved by their intended recipients. A user retrieves their e-mail messages from the mail
18 server by sending an inquiry message via the network to the mail server asking if there is
19 any mail stored there for them. Commonly these inquiry messages sent to the mail server
20 are known as "POP checks" because they check for mail according to the Post Office
21 Protocol. If the user's mail box is empty, then the mail server sends a negative response to
22 the user telling him so. On the other hand, if the mail server is storing mail messages for
23 the user, those messages are transmitted (in response to the POP check) to the user via the
24 network.

1 One problem with large data networks where the clients are connected at all times
2 without having to create a dial up connection is the large amount of network traffic due to
3 frequent POP checks that users make (or, more typically, that the users' computers makes
4 on the users' behalf) to see if there are any new e-mail messages waiting for them on the
5 mail server. The use of a POP3 mail system in a wide area network (WAN) may result in
6 a large amount of network traffic. This is not only a bandwidth problem, but also causes a
7 substantial loading on the servers across the network that have to handle and route all this
8 largely unproductive traffic. When a mail server is remote from the mail client, each
9 POP3 request may require numerous hops to transverse the network, and response must
10 travel the same distance.

11 It is largely unproductive traffic because the vast bulk of POP checks (over 90%,
12 typically) result in negative responses because POP checks are generated much more
13 frequently than the frequency with which e-mail messages arrive at the mail server. This
14 is very inefficient. Significant traffic handling server load reduction, and some bandwidth
15 savings, can be made if most POP checks are terminated close in the network to the
16 sender.

17 Thus, what is needed is a scheme for reducing the number of POP checks that are
18 transmitted over the network to the mail server.

19 Another e-mail related problem in networks is that the bandwidth demand resulting
20 from e-mail traffic is concentrated during certain times of the day. In particular, the
21 morning hours are a concentrated time for retrieval of e-mail messages from the mail
22 server. These e-mail-generated spikes in bandwidth demand present network management
23 challenges. One typical solution is to increase bandwidth capacity of the network to
24 accommodate demand spikes completely. Obviously, this is an expensive and inefficient

1 option because the added capacity will go largely unused (except in the event of demand
2 spikes). Another typical solution is to simply permit poor network performance during
3 periods when bandwidth demand spikes. Obviously, this option would be a source of
4 irritation to the users of the network.

5 Thus, what is needed is a scheme to accommodate e-mail retrieval traffic without
6 adding capacity that will go largely unused and without reducing network performance.

7 SUMMARY OF THE INVENTION

8 It is an object of the present invention to a scheme for reducing the number of POP
9 checks that are transmitted over the network to a POP mail server.

10 It is another object of the present invention to scheme to accommodate e-mail
11 retrieval traffic without adding capacity that will go largely unused and without reducing
12 network performance.

13 It is yet another object of the present invention to efficiently distribute e-mail
14 handled according to a POP system.

15 It is still another object of the present invention to push bandwidth demand and
16 traffic handling load toward the edges of a wide area network.

17 The present invention addresses this problem with two approaches that may be
18 used separately or together. The first is to attenuate the POP checks. The second is to
19 cache the e-mail messages.

20 Attenuation of POP checks is accomplished by intercepting each POP check packet
21 at a proxy server that is nominally local to where the user is located. The proxy server lets
22 the user's first POP check proceed on through the network to the mail server. Thereafter,
23 though, the proxy server only permits that user's received POP checks to proceed onward
24 according to a predetermined algorithm. For example, the proxy server may only permit a

1 POP check to proceed to the mail server if it has been at least fifteen minutes since the last
2 time the mail server was actually checked for e-mail by that particular user. When overly
3 frequent POP checks by that user are received prior to the permitted time, no actual check
4 of the mail server is permitted and the proxy server simply informs the user that he has no
5 mail (despite not knowing deterministically whether that is a true statement).

6 According to the cache aspect of the invention, a user's e-mail is cached at the
7 proxy server nearest to his presumed location. This decentralizes the e-mail storage away
8 from the mail server and spreads it out over the network at the various proxy servers. This
9 cache action is preferably done when there is a lull in network traffic (e.g., at night). This
10 can also be done as soon as mail is available so that the proxy deterministically knows that
11 a client has unread mail. This also has the effect of decentralizing the bandwidth demand
12 on the overall network since the e-mail messages have a shorter distance to travel when
13 retrieved by the user from the cache location at the proxy server.

14 **BRIEF DESCRIPTION OF THE DRAWINGS**

15 Additional objects and advantages of the present invention will be apparent in the
16 following detailed description read in conjunction with the accompanying drawing figures.

17 **Fig. 1** illustrates a network diagram implementing proxy handling of e-mail
18 according to an embodiment of the present invention.

19 **Fig. 2** illustrates a logical implementation for a subset of the signaling transactions
20 that take place between a client, a proxy server, and a mail server according to an
21 embodiment of the present invention.

22 **Fig. 3** illustrates a logical implementation for another subset of the signaling
23 transactions that take place between a client, a proxy server, and a mail server according to
24 an embodiment of the present invention.

1 **Fig. 4** illustrates a logical implementation for a subset of the signaling transactions
2 that take place between a client, a proxy server, and a mail server according to an alternate
3 embodiment of the present invention.

4 **Fig. 5** illustrates a logical implementation for a subset of the signaling transactions
5 that take place between a client, a proxy server, and a mail server according to a further
6 alternate embodiment of the present invention.

7 **Fig. 6** illustrates a logical implementation for a subset of the signaling transactions
8 that take place between a client, a proxy server, and a mail server according to another
9 alternate embodiment of the present invention.

10 **Fig. 7** illustrates a logical implementation for a subset of the signaling transactions
11 that take place between a client, a proxy server, and a mail server according to still another
12 alternate embodiment of the present invention.

13 **DETAILED DESCRIPTION OF THE INVENTION**

14 Typical e-mail client programs support a feature that performs automated mailbox
15 checking at configurable intervals. A user may set their e-mail program to query for new
16 mail every five minutes, for example. The “always on” characteristic of a broadband
17 Internet service enables a subscriber to leave their e-mail program running and
18 continuously polling for mail. This adds further to the load of POP3 packets on the
19 service provider’s network. And, while polling intervals will vary, empirical observation
20 shows that over 90% of the mailbox queries return no new mail. This problem of empty
21 POP checks is addressed by the attenuation aspect of the present invention.

22 Attenuation of POP checks is accomplished by a proxy server, which implements a
23 software proxy. The proxy server intercepts POP requests that originate with its locally
24 situated network component. As a proxy, it responds to a majority of these POP checks

1 based on its most recent knowledge of the state of the requestor's actual mailbox. The
2 proxy server uses an "attenuation interval" of n minutes. Thus, every n minutes it will
3 allow a mail query session (i.e., POP check) to flow through to the actual mail server. If
4 the mailbox is empty after this session, the proxy server will note that, and will return
5 "mailbox empty" responses to all POP checks for that mailbox for the next n minutes,
6 after which it will permit the subsequent POP check packet to flow through for that
7 mailbox, and so continue the pattern.

8 Generally speaking, the proxy server permits a user's first POP check to proceed
9 on through the network to the mail server. Thereafter, though, the proxy server only
10 permits that user's received POP checks to proceed onward according to a predetermined
11 algorithm. For example, the proxy server may only permit a POP check to proceed to the
12 mail server if it has been at least fifteen minutes since the last time the mail server was
13 actually checked for e-mail by that particular user. When overly frequent POP checks by
14 that user are received prior to the permitted time, no actual check of the mail server is
15 permitted and the proxy server simply informs the user that he has no mail (despite not
16 knowing deterministically whether that is a true statement).

17 While it is a worthy object to moderate the amount of traffic handling that is
18 required by the servers in a network, the attenuation solution should be implemented with
19 respect for the concerns of the e-mail users. Under certain circumstances, an e-mail user
20 may observe that the attenuation algorithm as described above may cause some delay in
21 how quickly they receive their e-mail messages. The user may perceive this not as an
22 optimization but, rather, as poor service. Accordingly, there is a need to accommodate the
23 expectations of the e-mail users to the extent possible.

1 One simple way to avoid user perception of delay service of e-mail messages is to
2 set the interval of n minutes at which POP check flow through is permitted as low as
3 possible. In fact, the value of n can be set dynamically, based on overall network load, so
4 that it is low when network load is light and set higher when network load peaks.

5 An optional feature of this attenuation process is that a custom SMTP extension
6 may provide an event service whereby the proxy server can receive notification when new
7 mail arrives for any mail account on the mail server for which it is acting as an attenuation
8 proxy. In case of such an event notification, it will permit the subsequent POP check
9 packet to flow through for that mailbox.

10 Another optional algorithm is to permit POP checks to flow through based upon
11 the number of POP checks that have been received for a given e-mail account. In other
12 words, only one out of every m POP checks is permitted to flow through to the mail
13 server. For obvious reasons, this is not a preferred method. It is mentioned simply
14 because of the ease with which it could be implemented.

15 The preferred attenuation algorithm is a combination of a time-based rule (i.e., wait
16 n minutes before letting another POP check through) and a demand-based rule (i.e., let the
17 next POP check through only if notice of actual mail receipt has been received from the
18 mail server). Although the demand-based rule alone may appear to be adequate, the
19 combination with a time-based rule ensures that e-mail still gets checked in the event that
20 the notification message from the mail server is not sent or fails to be routed to the proxy
21 server. The redundant use of the two rules provides a more robust system, and minimizes
22 the chances of causing user dissatisfaction with e-mail service.

23 The attenuation aspect of the present invention preferably makes a deterministic
24 assessment of whether an e-mail account being attenuated actually has any new e-mail

1 messages to be retrieved. One method is for the proxy server to snoop the actions of the
2 POP protocol and keep track of how many messages are read from the mail server and
3 how many were deleted from the mail server.

4 A simpler method to accomplish this is to wait until the e-mail client ends the e-
5 mail session. This happens when the e-mail client transmits a “quit” message. Rather
6 than letting the quit message flow through immediately, the proxy server temporarily
7 retains the quit message while it determines the status of that e-mail account. While the
8 quit command is being held, the proxy server sends a “stat” command to the mail server
9 asking, in essence, “Do you have mail for me?” If the answer received from the mail
10 server is “no,” then the proxy server knows deterministically that the e-mail client has no
11 un-retrieved messages. This status is cached locally at the proxy server. Subsequently,
12 the quit message is permitted to travel onward through the network to the mail server.

13 One algorithm for implementing the attenuation aspect of the present invention
14 uses a set of tables. When a new user checks their e-mail account on the mail server (i.e.,
15 sends an initial POP check), the proxy server places that user in an attenuation table.
16 Assuming that the state of the user’s mailbox is determined, a timer is started when the
17 user is placed in the table. The user remains in the table until the timer times out. Each
18 time the user transmits a POP check, their POP check is attenuated at the proxy server if
19 the user still remains listed in the table. In other words, until the timer times out and
20 removes the user from the attenuation table, that user’s mail client software will continue
21 to receive pseudo responses to its POP checks telling the user that the mail server has no
22 mail for that account.

1 Another triggering event for removing the user from the attenuation table is if a
2 notification message is received from the e-mail server that the user has mail. By being
3 removed from the table, the user's next POP check is passed through to the mail server.

4 Every time the user is removed from the attenuation table (for whatever reason:
5 time out, notification, server reset, etc.) that user's next POP check is permitted to flow
6 through to the mail server. Concurrently, the transmission of that POP check places the
7 user back into the attenuation table and re-starts the timer. In this way, the user's own
8 actions re-establish the state in an attenuation queue.

9 In forming the tables in this implementation, the username information is cached.
10 To avoid the need to perform linear searches, a hash table algorithm is used. Each hash
11 entry in the table is a fixed length value, e.g., a five hash, created according one of plural
12 hash algorithms that are well known in the database art. These hashes are used as keys for
13 entry in a hash table. The first time a POP check is seen from a user, that user's username
14 is hashed and entered as an entry in the hash table. The hash table uses the key-value pair.
15 The key is the hash, and the value is the pointer to a small data structure somewhere in the
16 proxy server's memory. The data structure contains the username in clear text, as well as
17 the user's password in clear text. The data structure also may contain information
18 indicating the state of the user's mailbox, and a value indicating the elapsed time left for
19 that user.

20 This implementation also uses another table, which is an ordered table that
21 indicates which one of the hash values (indicative of the individual users) will expire next.
22 The ones that will expire soonest are placed at the top of the ordered table, and the ones
23 that will expire last are placed at the bottom of the ordered table. The ordered table may
24 be thought of as a timer list.

1 Thus, when a user first checks for e-mail, their username is hashed and the hash is
2 placed both in a hash table and at the end of the timer list ordered table. An algorithm
3 then regularly checks the timer list to see if one or more of the hashes at the top of the list
4 have timed out.

5 A more detailed discussion of this preferred implementation is discussed as
6 follows.

7 The attenuation algorithm has two main logical components -- a table of users and
8 a finite state machine. The finite state machine represents the logic that maps to any
9 particular combination of user state and POP3 command.

10 The user table is a hash table where the hash key is the user's mailbox account
11 name for the POP3 server that the proxy server is proxying. A user's entry in the table has
12 attributes, including mailbox password (may be encrypted or plain text), a time stamp of
13 the last time the mailbox was verified as empty, state (where the state is the user's current
14 state in the sequence of commands that make up a POP3 session), lock flag, terminate
15 flag, timeout field, etc.

16 **States, Events, and Actions**

17 What follows is a list of possible states for a POP3 attenuation session, and the
18 actions and state transitions that result from various events, where the main event type is a
19 POP3 command. A session associates a command with a mailbox username/usertable
20 entry; although the POP3 **user** command is the only command that carries that name, an
21 implementation mechanism can make this association for subsequent commands within
22 the session.

23 None State. Event: POP3 proxy server receives the **user** command, but there is no
24 corresponding entry in the user table. The POP3 proxy server creates a table entry for this

1 user, forwards the command to the POP3 mail server, and returns the response to the
2 client. The state transitions to Authorization.

3 Idle State. Event: POP3 proxy server receives the **user** command and finds a
4 corresponding entry in the table. Sends +OK response to the client. The state transitions
5 to Authorization.

6 Authorization State. Event: The POP3 proxy server receives the **pass** command.

- 7 • If the Lock flag equals true, send response: -ERR Your mail box is locked
8 by another POP3 session. The state remains as Authorization.
- 9 • If the password sent equals the password attribute in the user table entry
10 and the time elapsed since the last-time-mailbox-empty time stamp is less
11 than the global attenuation interval, set the terminate flag to true.
12 Additionally, set the Lock flag to true, start the Session Timer to wait for
13 possible timeout, and send a +OK response to the client. The state
14 transitions to Transaction.
- 15 • If the password sent equals the password attribute in the user table entry
16 and the time elapsed is greater than the global attenuation interval, set the
17 Terminate flag to false. Additionally, send the **user** command to the POP3
18 mail server, wait for a response, send the **pass** command, and wait for a
19 response. Return the response to the client. If the response from the server
20 is -ERR, the state does not change. If the response is +OK, the state
21 transitions to Transaction.
- 22 • If the password sent does not equal the password attribute in the user table
23 entry, send the **user** command to the POP3 server, wait for a response, send
24 the **pass** command, and wait for a response. Return the response to client.

1 IF the response from the server is –ERR, the state does not change. If the
2 response is +OK, save the new password; the state transitions to
3 Transaction.

4 Authorization State. Event: The POP3 proxy server receives the **quit** command.
5 The Lock flag is set to false. The state transitions to idle.

6 Transaction State. Event: The POP3 proxy server receives any of these commands:

- 7 • **stat** -- If the terminate flag equals true, the POP3 proxy server returns +OK
8 0 0. This tells the client that their mailbox is empty. If the terminate flag
9 equals false, the POP3 proxy server forwards the request to the POP3 mail
10 server and returns the response to the client. The state remains as
11 Transaction.
- 12 • **list** -- If the terminate flag equals true, the POP3 proxy server returns –ERR
13 no such message. If the terminate flag equals false, the POP3 proxy server
14 forwards the request to the POP3 mail server and returns the response to the
15 client. The state remains as Transaction.
- 16 • **retr** -- If the terminate flag equals true, the POP3 proxy server returns –
17 ERR no such message. If the terminate flag equals false, the POP3 proxy
18 server forwards the request to the POP3 mail server and returns the
19 response to the client. The state remains as Transaction.
- 20 • **dele** -- If the terminate flag equals true, the POP3 proxy server returns –
21 ERR no such message. If the terminate flag equals false, the POP3 proxy
22 server forwards the request to the POP3 mail server and returns the
23 response to the client. The state remains as Transaction.

- 1 • **noop** -- The POP3 proxy server returns +OK. The state remains as
2 Transition.
- 3 • **rset** -- The POP3 proxy server returns +OK. The state remains as
4 Transition.
- 5 • **top** -- If the terminate flag equals true, the POP3 proxy server returns –
6 ERR no such message. If the terminate flag equals false, the POP3 proxy
7 server forwards the request to the POP3 mail server and returns the
8 response to the client. The state remains as Transaction.
- 9 • **uidl** -- If the terminate flag equals true, the POP3 proxy server returns –
10 ERR no such message. If the terminate flag equals false, the POP3 proxy
11 server forwards the request to the POP3 mail server and returns the
12 response to the client. The state remains as Transaction.
- 13 • **quit** -- If the terminate flag equals true, the POP3 proxy server returns
14 +OK. The Lock flag is set to false, and the state transitions to Idle. If the
15 terminate flag equals false, the POP3 proxy server sends a stat command to
16 the POP3 mail server, to retrieve the number of messages now in the
17 mailbox. If the number is 0, the program sets the last-time-mailbox-empty
18 time stamp to the current time; else it sets the time stamp to NULL. Then it
19 sends quit to the POP3 mail server, and returns the result to the client. The
20 Lock flag is set to false, and the state transitions to Idle.

21 Transaction State. Event: The POP3 Attenuation Session Timeout Event. The
22 timer popped for a locally terminated session. The POP3 proxy server ends the session
23 connection. The Lock flag is set to false, and the state transitions to Idle.

Garbage Collection

The proxy server performs a garbage collection function. That is, at configurable intervals, or in case of need, it may delete all inactive entries in its user table. A case of need is defined, for example, as a user table that has become full. Other definitions are possible.

After the garbage collection, the proxy server can rebuild its user table, as new POP checks come in.

Opt Out List

The proxy server may optionally support an Opt Out List. This is simply a list of user mailbox account names that won't be subject to proxy treatment. All POP3 commands for names on this list are passed through to the POP3 mail server.

The Fast Check Case

The proxy server may optionally support "fast check" logic to account for a case where a user knows that he has new mail (through some non-POP3 source of knowledge, as for example, when has just sent an e-mail to himself), and where he does rapid and repeated new mail checks within a short interval, to get the mail as soon as possible.

The basic function of the "fast check" logic is to keep track of the time of each new mail query of a user. If a certain number, say three checks, came through in a short interval, such as 30 seconds, the third check could be permitted through to the mail server. This would reward the user's persistence. However, if the user continued to check rapidly after the third (where third is used as one example of a configurable value) check, subsequent checks would be terminated locally until another, longer, interval had expired. This would limit the effect of excessive, automated, or potentially malicious querying.

1 One way to discriminate the difference between fast checks that are the product of
2 persistent manual checking by the user and automated checks where the check interval has
3 been set unreasonably low (e.g., one every five seconds) is to assess the periodicity of the
4 checks. If the time interval T between the quickly repeated checks is metronomically
5 regular, then it is adjudged to be nothing more than automated checking. On the other
6 hand, if the time interval T varies substantially, then it is adjudged to be manual fast
7 checking. Some permissiveness is allowed for rewarding the persistence of a user who is
8 fast checking manually. However, fast automated checks are attenuated ruthlessly.

9 Referring to **Fig. 2**, a logical implementation for a subset of the signaling
10 transactions that take place between a client, a proxy server, and a mail server is
11 illustrated. The signal transactions shown are appropriate for the case where the mail
12 client does not already exist in the table.

13 Referring to **Fig. 3**, a logical implementation for another subset of the signaling
14 transactions that take place between a client, a proxy server, and a mail server is
15 illustrated. The signal transactions shown represent the case where the mail client is being
16 attenuated (i.e., is in the table).

17 Referring to **Fig. 4**, a logical implementation for yet another subset of the signaling
18 transactions that take place between a client, a proxy server, and a mail server is
19 illustrated, according to an embodiment of the present invention. The signal transaction
20 shown handles the situation where the client has changed their password.

21 Referring to **Fig. 5**, a logical implementation for still another subset of the
22 signaling transactions that take place between a client, a proxy server, and a mail server is
23 illustrated, according to an embodiment of the present invention. The signal transaction
24 shown handles the situation where an erroneous password is used.

1 Referring to **Fig. 6**, a logical implementation for an additional subset of the
2 signaling transactions that take place between a client, a proxy server, and a mail server is
3 illustrated. The signal transaction illustrated implements synchronization for cache in
4 transaction state (wherein messages have been cached).

5 Referring to **Fig. 7**, a logical implementation for a further subset of the signaling
6 transactions that take place between a client, a proxy server, and a mail server according to
7 an embodiment of the present invention. Possible synchronization for cache in transaction
8 state (messages have been cached)

9 Preferably, the present invention is implemented according to soft state principles.
10 Each proxy server should be able to build up its state from the information around it,
11 without need for extensive backing up of previous state information. From an initial state
12 of not knowing anything about user's or their accounts, the proxy server gradually learns
13 and builds a knowledge base, using only the algorithms it is programmed to implement
14 and observation of the POP mail processes that occur around it.

15 For example, when a first POP check for an e-mail account is received by the
16 proxy server (assuming it is starting from a no memory state), the proxy server permits the
17 POP check to flow through to the mail server and starts to build a list. The proxy server
18 determines the state of the mailbox and then keeps track of the state information. As this
19 process repeats for different users, the proxy server builds up a table of users and the state
20 of their respective mailboxes. This self-teaching aspect of soft state operation eliminates
21 the need for expensive state back-up resources to store state information for later retrieval
22 in the event of failure of the proxy server. A dead proxy server may simply be replaced by
23 a similar machine what will teach itself what it needs to know.

1 E-mail traffic load varies with time. There is little that can be done to control
2 when e-mail messages are transmitted to a POP mail server. However, there is some
3 opportunity to manipulate the traffic patterns for how and when e-mail is distributed from
4 the mail server to the intended recipients. This may be exploited to minimize peak traffic
5 loads to the extent that the load is caused by retrieval of e-mail messages.

6 According to the cache aspect of the invention, a user's e-mail is cached at the
7 proxy server nearest to his presumed location. This decentralizes the e-mail storage away
8 from the mail server and spreads it out over the network at the various proxy servers. This
9 cache action is preferably done when there is a lull in network traffic (e.g., at night). This
10 also has the effect of decentralizing the bandwidth demand on the overall network since
11 the e-mail messages have a shorter distance to travel when retrieved by the user from the
12 cache location at the proxy server.

13 By this cache action, the user's mail is moved out to the edges of the network
14 ahead of when the user will be seeking to retrieve it. In a sense, the mail messages are
15 pushed to the user, at least part of the way, to a proxy server near where it is presumed that
16 the user will connect to the network when he seeks to retrieve his e-mail. This
17 presumption will not always be accurate, however, since users are free to connect to the
18 network at geographically diverse points (e.g., when they are traveling). More often than
19 not, though, the presumption should be accurate.

20 One algorithm for choosing which proxy server to push a given user's e-mail to is
21 to always assume that the same proxy server (entered in the user's profile as their "home"
22 proxy server) will be appropriate. This algorithm does not adapt to a user's changing
23 position. This algorithm relies only upon a default user profile as a reference, not a
24 dynamic state table.

1 Another algorithm for choosing which proxy server to push a given user's e-mail
2 to is to select the proxy server that is the most proximate to the last known point on the
3 network where the user accessed the network. This algorithm is dynamic and adapts to a
4 user's changing geographic position. However, this algorithm is not completely consistent
5 with a soft state approach where there is no reliance upon keeping a memory of previous
6 state information. In the event of a loss of state information, the system may default to the
7 algorithm described in the preceding paragraph.

8 In the event that the mail cache algorithm is wrong and sends email to the wrong
9 proxy server to be cached, the user logging onto the network at another place (and
10 interacting via an entirely different proxy server) may still retrieve the messages from the
11 mail server. That is because the mail server does not delete the messages from its memory
12 when it pushes the messages out to a selected proxy server. Until retrieved (from one
13 server or another), the messages are redundantly stored on both servers simultaneously.

14 This redundant storage of e-mail messages raises issues of possible
15 unsynchronized states. An unsynchronized state occurs when a user has read and deleted
16 some messages from one server, but redundant copies of those messages remain stored on
17 another server. This could result in a confusing situation where the user will retrieve a
18 message that has already been retrieved and deleted in a previous e-mail session. To avoid
19 such confusing occurrences, a synchronization algorithm may be used.

20 A synchronization algorithm according to one embodiment of the present invention
21 makes use of unique identifiers (UIDs) that are assigned to each message. When e-mail
22 messages are retrieved from a server (either from a proxy server or from the centralized
23 mail server), a synchronization handshake occurs between the mail server and the relevant
24 proxy server.

1 For example, if cached messages are retrieved from a proxy server, a
2 synchronization inquiry is sent from the proxy server to the mail server inquiring whether
3 the mail server contains messages in that user's mail box with the UIDs that match those
4 of messages that the user has just retrieved locally. If there is identity of UIDs between
5 the proxy server's cache mail box and the mail server's central mail box, then the user is
6 served the locally cached copies of the messages. When the user deletes a message, that
7 message is deleted from both servers virtually simultaneously (allowing for transmission
8 delay across the network). However, if there is a difference between the UIDs of
9 messages residing on the two servers, the locally cached messages are thrown away and
10 the messages stored at the mail server are retrieved for the user. In this manner, any
11 confusion or conflict is avoided.

12 Referring to **Fig. 1**, a network diagram illustrates how proxy handling of e-mail
13 may be implemented according to an embodiment of the present invention. An e-mail
14 message originates from an originator client **20**, addressed to the e-mail account of an
15 intended recipient client **40**. The e-mail message travels from the originator **20**, via a
16 network **10**, to an e-mail server **30**. The process by which the e-mail message travels from
17 the mail server **30** to the recipient **40** follows one of two general scenarios.

18 In the event that the recipient **40** is not operating their e-mail client software
19 contemporaneously with the receipt of the e-mail message by the mail server **30**, the e-
20 mail message may be cached. When traffic load on the network ebbs, the e-mail message
21 is cached at a proxy server **50**, which is nominally local to the recipient **40**. When the
22 recipient **40** subsequently initiates their e-mail client software, the recipient transmits a
23 POP check (via the network **10**) that is intercepted by the proxy server **50**. In reply to the
24 POP check, the proxy server **50** transmits to the recipient **40** the cached e-mail message

1 and informs the e-mail server **30** that that message may be deleted from the e-mail server's
2 memory.

3 In the event that the recipient **40** is operating its e-mail client software
4 contemporaneously with the receipt of the e-mail message by the mail server **30**, the e-
5 mail message is retrieved from the e-mail server **30** subject to any attenuation activity that
6 the proxy server **50** may perform. When the proxy server **50** permits a POP check from
7 the recipient to pass through to the e-mail server **30**, the e-mail server **30** transmits the e-
8 mail message over the network **10** directly to the recipient **40**.

9 The present invention has been described in terms of preferred embodiments,
10 however, it will be appreciated that various modifications and improvements may be made
11 to the described embodiments without departing from the scope of the invention. The
12 scope of the present invention is limited only by the appended claims.

WHAT IS CLAIMED IS:

1 1. A method of moderating traffic load on network servers in a network where
2 electronic mail is retained for retrieval from at least one mail server, the method
3 comprising:
4 permitting a mail request for a mail client to pass through a proxy server to the
5 mail server; and
6 attenuating subsequent mail requests for the mail client at the proxy server until a
7 predetermined condition has been satisfied.

1 2. The method of claim 1, wherein the predetermined condition is a predetermined
2 period of time.

1 3. The method of claim 2, wherein the predetermined period of time is
2 dynamically determined based on the amount of traffic load on the network.

1 4. The method of claim 1, wherein the predetermined condition is a combination
2 of a predetermined time period and receipt of a notification from the mail server that mail
3 has been received for the mail client at the mail server, whichever occurs first.

1 5. The method of claim 4, wherein the predetermined period of time is
2 dynamically determined based on the amount of traffic load on the network.

1 6. The method of claim 1, wherein attenuating subsequent mail requests is
2 suspended in the event it is determined that a user is manually initiating rapidly repeated
3 mail requests.

1 7. The method of claim 1, wherein attenuating includes blocking the subsequent
2 mail requests from transmission across the network to the mail server.

1 8. The method of claim 1, wherein the predetermined condition is independent of
2 time.

1 9. A method of managing bandwidth usage in a network where electronic mail is
2 retained for retrieval from at least one mail server, the method comprising:

3 selecting a time when network bandwidth load is low; and
4 pushing unretrieved mail messages to a proxy server at the selected time, wherein
5 the pushed mail messages are cached at the proxy server.

1 10. The method of managing bandwidth usage of claim 9, wherein selecting a time
2 is based on when bandwidth load at a predetermined point in the network falls below a
3 predetermined threshold.

1 11. The method of managing bandwidth usage of claim 9, wherein selecting a time
2 is based on a predetermined time of day.

1 12. A proxy server for use in a network where electronic mail is retained for
2 retrieval from at least one mail server, the proxy server comprising:

3 a processor, and
4 a memory including software instructions adapted to enable the proxy server to
5 perform the steps of:
6 permitting a mail request for a mail client to pass through the proxy server to
7 the mail server; and

8 attenuating subsequent mail requests for the mail client at the proxy server
9 until a predetermined condition has been satisfied.

1 13. The proxy server of claim 12, wherein the predetermined condition is a
2 predetermined period of time.

1 14. The proxy server of claim 13, wherein the predetermined period of time is
2 dynamically determined based on the amount of traffic load on the network.

1 15. The proxy server of claim 12, wherein the predetermined condition is a
2 combination of a predetermined time period and receipt of a notification from the mail
3 server that mail has been received for the mail client at the mail server, whichever occurs
4 first.

1 16. The proxy server of claim 15, wherein the predetermined period of time is
2 dynamically determined based on the amount of traffic load on the network.

1 17. The proxy server of claim 12, wherein attenuating subsequent mail requests is
2 suspended in the event it is determined that a user is manually initiating rapidly repeated
3 mail requests.

1 18. The proxy server of claim 12, wherein attenuating includes blocking the
2 subsequent mail requests from transmission across the network to the mail server.

1 19. The proxy server of claim 12, wherein the predetermined condition is
2 independent of time.

1 20. A mail server for use in a network where electronic mail is retained for
2 retrieval from the mail server, the mail server comprising:

3 a processor, and
4 a memory including software instructions adapted to enable the proxy server to
5 perform the steps of:

6 selecting a time when network bandwidth load is low; and

7 pushing unretrieved mail messages to a proxy server at the selected time,

8 wherein the pushed mail messages are cached at the proxy server.

1 21. The mail server of claim 20, wherein selecting a time is based on when
2 bandwidth load at a predetermined point in the network falls below a predetermined
3 threshold.

1 22. The mail server of claim 20, wherein selecting a time is based on at a
2 predetermined time of day.

1 23. A network comprising:
2 at least one mail server where electronic mail is retained for retrieval by mail
3 clients;

4 a plurality of proxy servers distributed about the network;

5 wherein the mail server caches unretrieved mail messages at the proxy servers.

1 24. The network of claim 23, wherein unretrieved mail messages are cached at a
2 selected time.

1 25. The network of claim 24, wherein the selected time is determined to be when
2 bandwidth load at a predetermined point in the network falls below a predetermined
3 threshold.

1 26. The network of claim 24, wherein the selected time is a predetermined time of
2 day.

1 27. The network of claim 23, wherein the mail server synchronizes with the
2 plurality of proxy servers periodically to ensure that when changes are made to a message
3 on the mail server or on the proxy server that the changes are reconciled.

1 ~~28.~~ A network comprising:
2 at least one mail server where electronic mail is retained for retrieval by mail
3 clients;
4 a plurality of proxy servers distributed about the network;
5 wherein each of the proxy servers comprises:
6 a processor, and
7 a memory including software instructions adapted to enable the proxy
8 server to perform the steps of:
9 permitting a mail request for a mail client to pass through the proxy
10 server to the mail server; and
11 attenuating subsequent mail requests for the mail client at the proxy
12 server until a predetermined condition has been satisfied.

1 ~~29.~~ A network comprising:
2 a mail server where electronic mail is retained for retrieval by mail clients;

3 a plurality of proxy servers distributed about the network;

4 wherein the mail server comprises:

5 a processor, and

6 a memory including software instructions adapted to enable the mail server

7 to perform the steps of:

8 selecting a time when network bandwidth load is low; and

9 pushing unretrieved mail messages to a proxy server at the selected

10 time, wherein the pushed mail messages are cached at the proxy

11 server.

1 30. The network of claim 29, wherein selecting a time is based on when
2 bandwidth load at a predetermined point in the network falls below a predetermined
3 threshold.

1 31. The network of claim 29, wherein selecting a time is based on at a
2 predetermined time of day.

ABSTRACT OF THE DISCLOSURE

1 Traffic handling load on network servers is moderated by attenuating POP checks
2 at proxy servers located across the network. Attenuation of POP checks is accomplished
3 by intercepting each POP check packet at a proxy server that is nominally local to where
4 the user is located. The proxy server permits a given user's initial POP check to proceed
5 on through the network to the mail server. Thereafter, though, the proxy server only
6 permits that user's received POP checks to proceed onward according to a predetermined
7 algorithm, e.g., at intervals of no less than fifteen minutes. Overly frequent POP checks
8 by a user are responded to by the proxy server (rather than the mail server itself) with a
9 response indicating that the user that he has no mail (despite not knowing deterministically
10 whether that is a true statement). Additionally the proxy server may buffer and delay the
11 POP transactions to effectively allow only a predetermined rate of POP checks.
12 Bandwidth loading on the network is managed by pushing e-mail message traffic out to
13 the edges of the network at times when bandwidth demand is low. To accomplish this, a
14 user's e-mail is cached at the proxy server nearest to his presumed location. This
15 decentralizes the e-mail storage away from the mail server and spreads it out over the
16 network at the various proxy servers. This cache action is preferably done when there is a
17 lull in network traffic (e.g., at night). This has the effect of decentralizing the bandwidth
18 demand on the overall network since the e-mail messages have a shorter distance to travel
19 when retrieved by the user from the cache location at the proxy server.

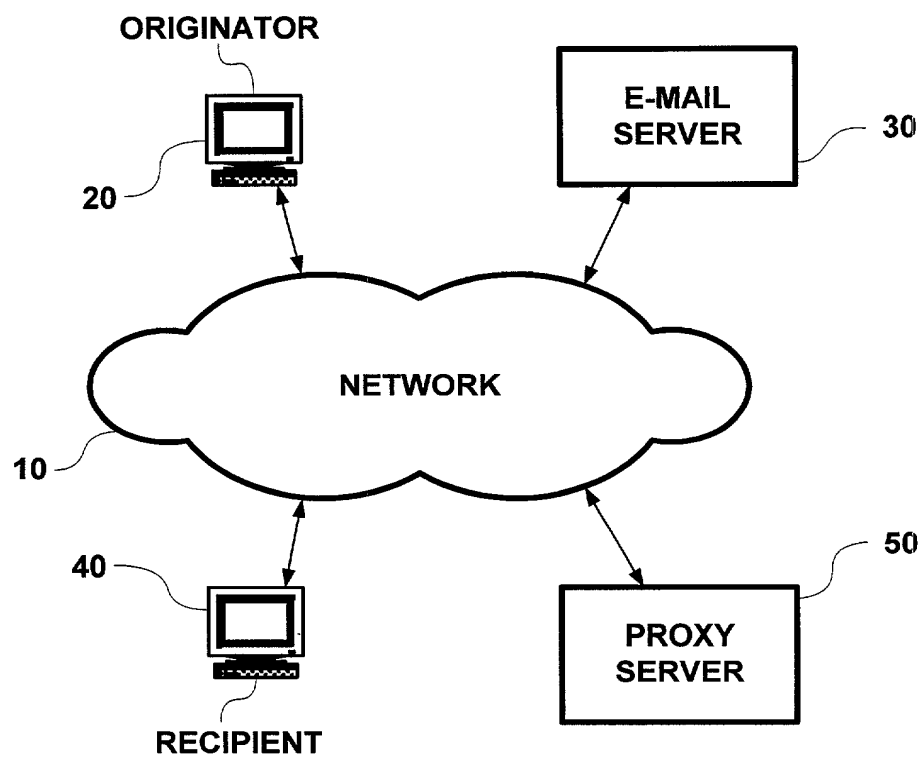


FIG. 1

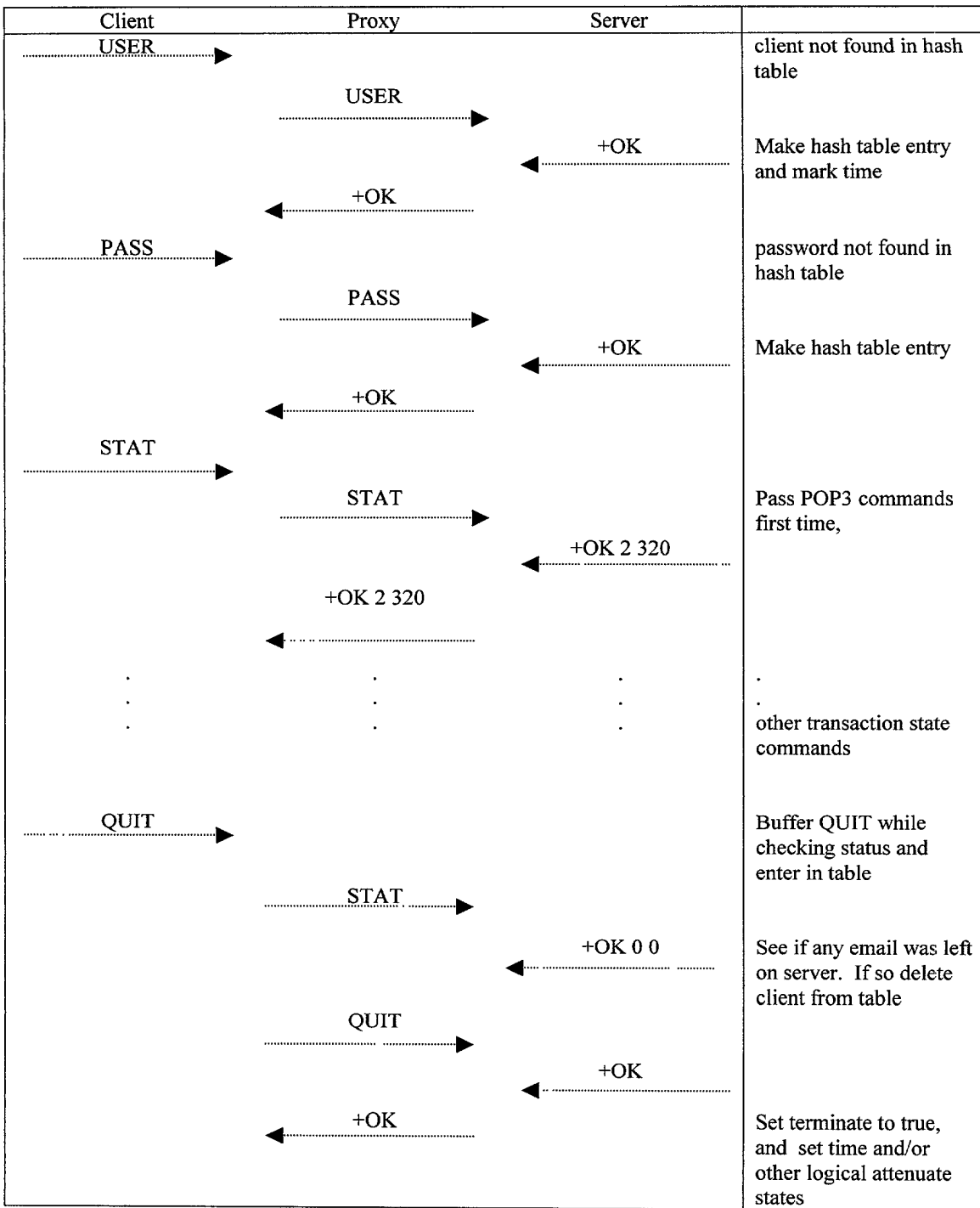


FIG. 2

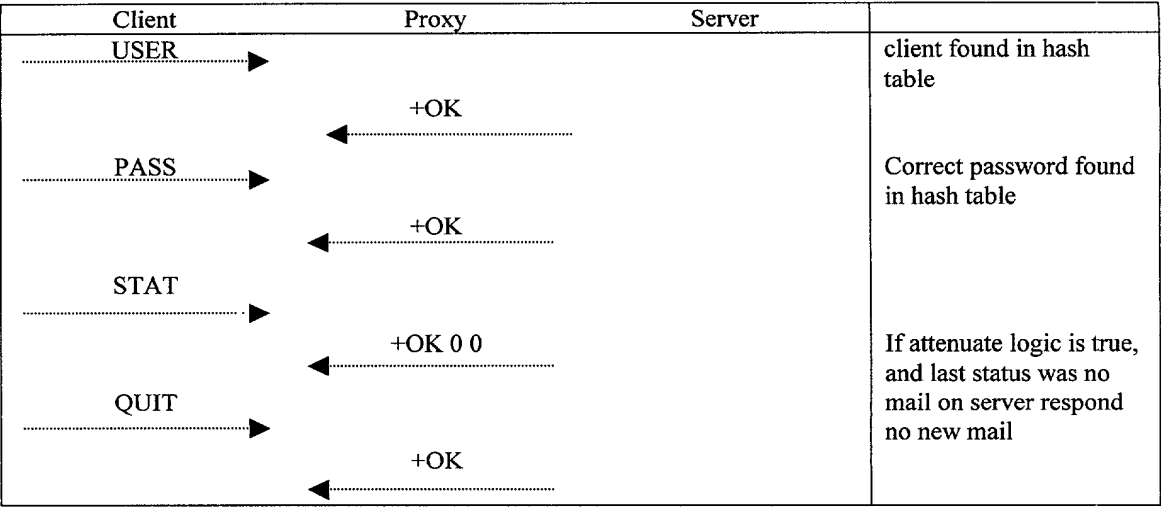


FIG. 3

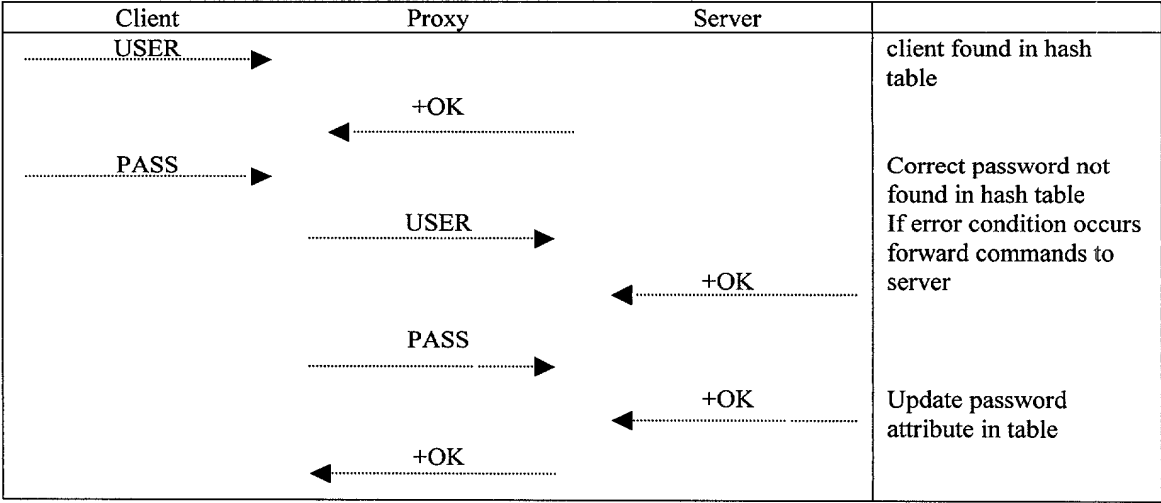


FIG. 4

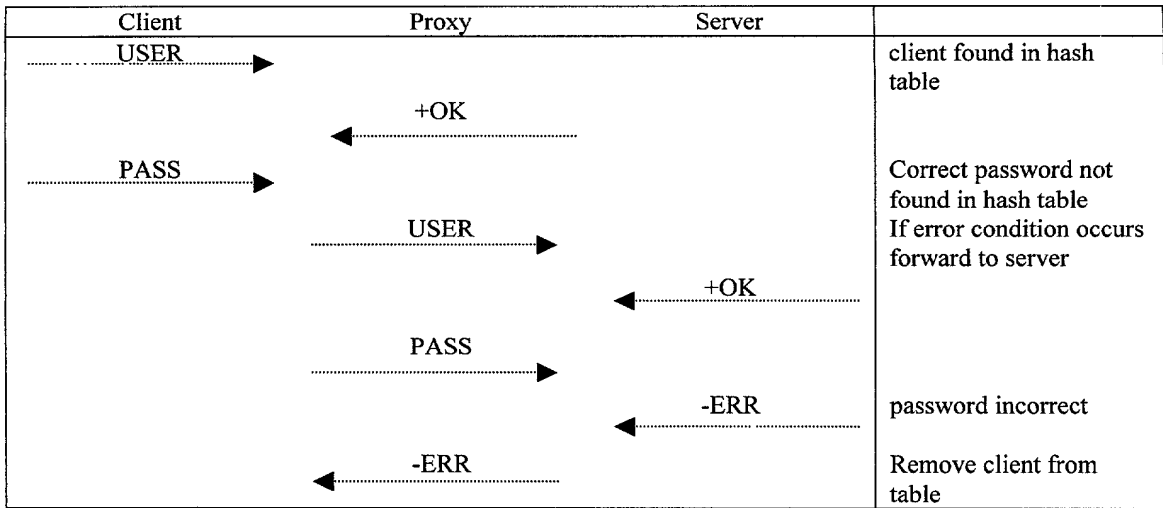


FIG. 5

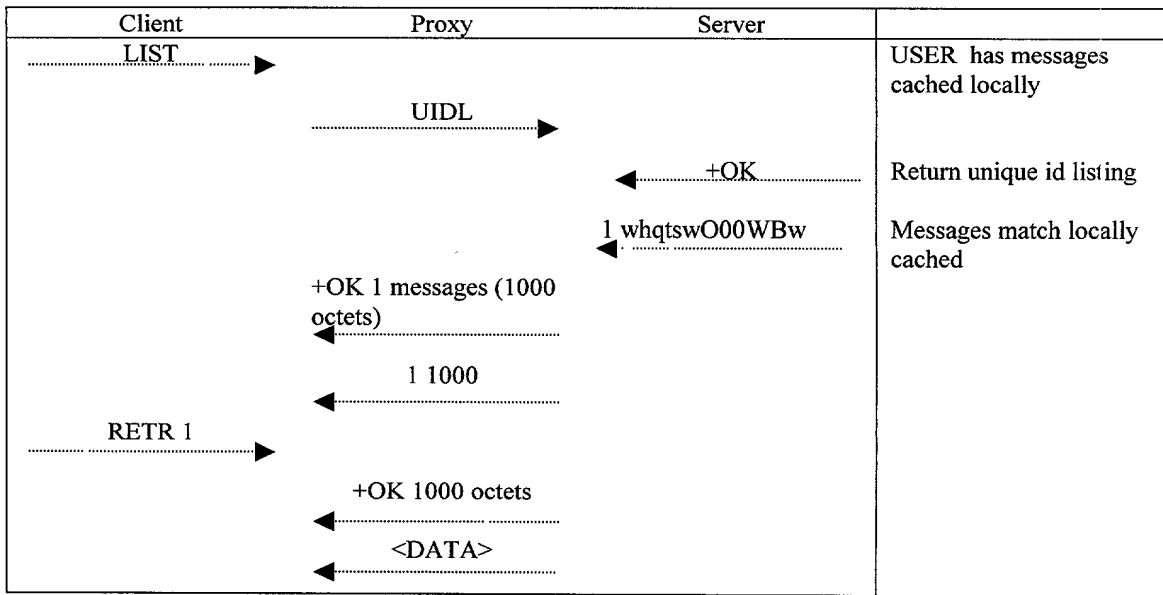


FIG. 6

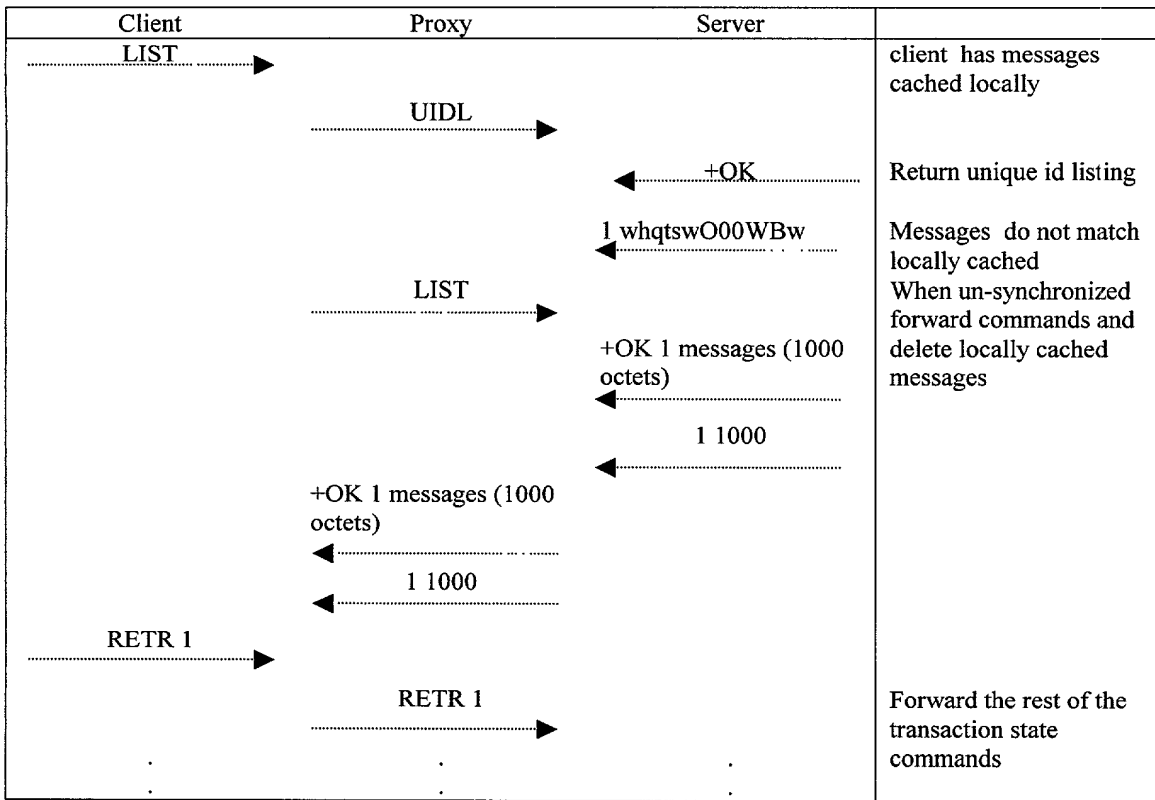


FIG. 7

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of Howard Pfeffer and John Leddy

Serial No.: Not Yet Assigned

Group Art Unit:

Filed: HERewith

Examiner:

FOR: REDUCTION OF NETWORK SERVER LOADING

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As below inventors, we hereby declare that:

Our residence, post office address and citizenship are as stated below next to our names.

We believe we are the original, joint and first inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled **REDUCTION OF NETWORK SERVER LOADING**, the specification of which is attached hereto.

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, § 1.56(a).

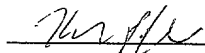
We hereby appoint the following attorney(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Jon L. Roberts
Registration No. 31,293
John K. Abokhair, Esq.
Registration No. 30,537
Kevin L. Pontius
Registration No. 37,512
Christopher B. Kilner
Registration No. 45,381
Roberts Abokhair & Mardula, L.L.C.
11800 Sunrise Valley Drive
Suite 1000

Reston, Virginia 20191-5302
(703) 391-2900

We declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of inventor Howard Pfeffer

Inventor's Signature  Date 1/27/2000

Residence Virginia

Post Office Address: 12021 Creek Bend Drive; Reston VA 20194

Citizenship U.S.A.

Full name of inventor John Leddy

Inventor's Signature  Date 3/22/2000

Residence Virginia

Post Office Address: 1247 Lamplighter Way; Reston VA 20171

Citizenship U.S.A.